Controlling a True or False Quiz Through Eye Movement

Max Xiao (480377535) and Tyrone Simbul (460379786)

School of Physics, University of Sydney, NSW 2006, Australia

Compiled May 3, 2022

Our project was the construction of a trivia quiz that can be played with just eye-movements - which required a complex synthesis of both Data Science and Physics concepts. This report will address the method by which we created it, explain the key underlying concepts and discuss areas for improvement.

1. MOTIVATION FOR PRODUCT

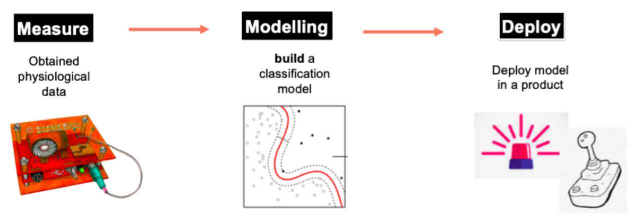

Fig. 1. Product: from input to output [15].

Our aim is to produce a Trivia Game that can be played with eye-movements like figure 1. In order to do that, our team has synthesised a combination of Physics and Data Science concepts to create a product which can interpret eye movements. In the broader context, we believe our project has significant use cases in society. For example, by gamifying the movement of the eye - it has application as a training tool for patients recovering from brain damage [16]. Further, for enthusiastic gamers, by removing the need for a key-board and mouse our product allows people to play the game in spine positions healthier for our posture. Extending beyond our product, the uses for our core technology which detects eye movements can be applied in an even wider set of settings - such as changing the television channel.

2. GAME PRODUCT DESCRIPTION

In our prototype game a player will have 2 minutes to answer as many questions as possible. In order to construct a feasible game, we are looking to implement the controls:

- Looking left translates to true
- Looking right translates to false
- Blinking translates to a skip

When the 120 seconds are finished, a message will be presented to the player notifying them of their score and the number of questions skipped:

"You scored 4 out of 7! Skipped : 2"

3. KEY CONCEPTS

A. Physics

A.1. Spikerbox

In this project, we used a Backyard Brains Spikerbox which relies on Electro-oculography (EOG). This is the measurement and analysis of eye movements based on the potential difference between the cornea and the Bruch membrane seen in figure 2. That potential is called the standing potential [4].

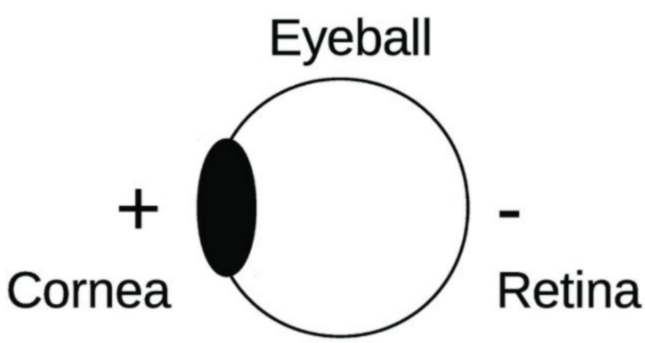

Fig. 2. The cornea relative to the retina has a net positive charge. Bruch's membrane is the innermost layer covering the retina. Thus, an electric field pointing to the right would exist [3].

Movements of the eye relative to the electrodes on the headband in figure 3 will produce an electrical signal. This causes a change in the direction of the electric field within the eye. This encourages charged particles to move and equilibrate themselves. The change in potential from the standing potential is what is measured and an example can be seen in figure 4. EOG recordings of eye movement are precise and have been demonstrated to be capable of being accurate within 0.5° and can perceive deviations in eye movement of 0.1° [1].

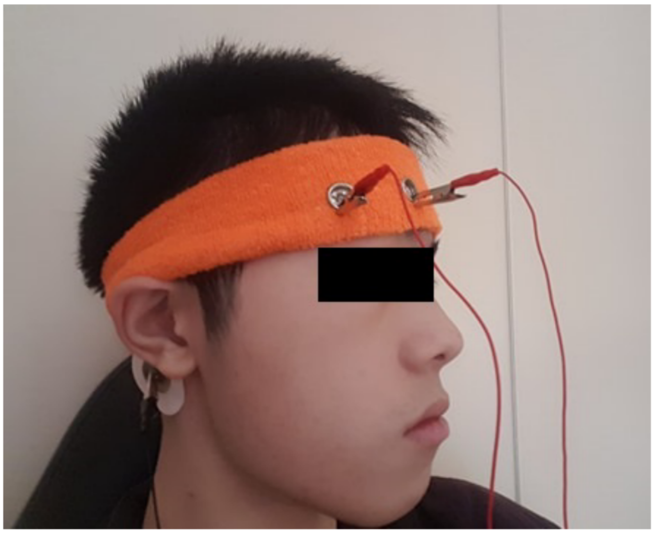

Fig. 3. Proposed EOG setup are (a) conventional wet electrode placed behind the ear. (b) Wireless EOG acquisition system contains a preamplifier, a filter, a microcontroller, and a wireless module. (c) Headband with 2 electrodes which have either eye (left or right) centred in between them.

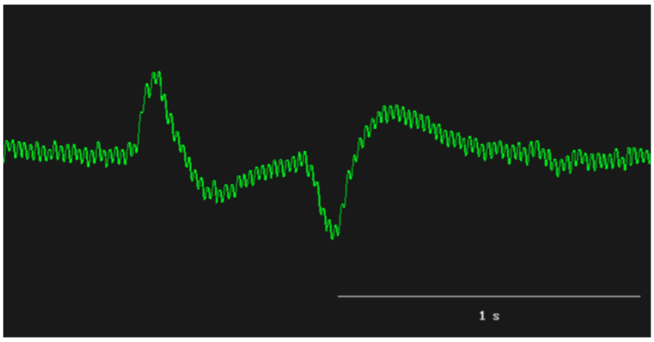

Fig. 4. The x-direction represents time, while the y-axis represents potential relative to resting potential. The amplitude of signal will increase if electrode gel is applied before putting on the equipment. This signal corresponds with a left look which quickly realigns back to looking straight.

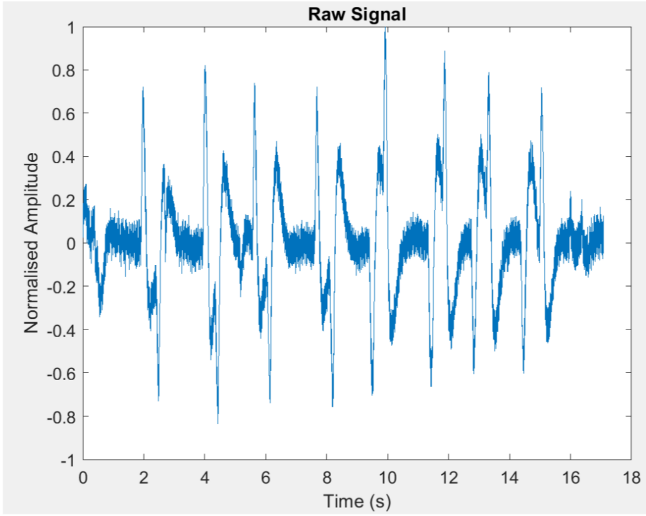

Fig. 5. Uncleaned normalised signal measured.

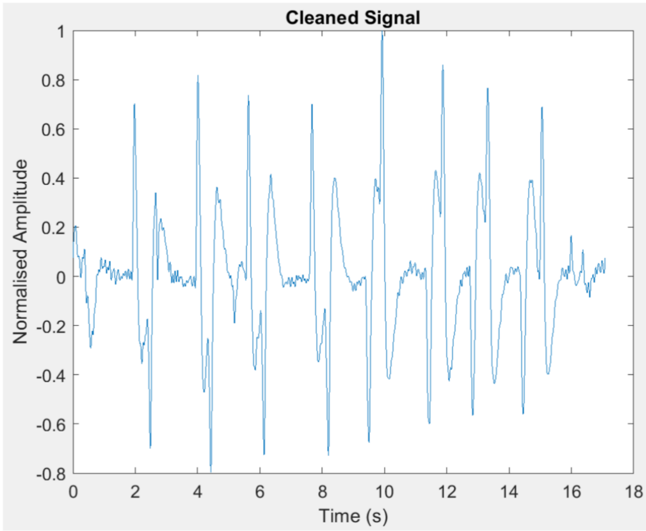

Fig. 6. Cleaned signal with frequencies higher than 15 Hz filtered out.

A.2. Fourier Transform

One of the most ground-breaking concepts in recent Mathematical history is the idea of Fourier Transform. It is the concept that any function can be represented using a linear combination of sine and cosine waves. Using Euler's formula in equation 1, complex exponential form is used to simplify the expression of sine and cosines.

$$e^{i\theta} = \cos\theta + i\sin\theta \tag{1}$$

$$c_n = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(x)e^{-2\pi i \frac{n}{T}x}\,dx \tag{2}$$

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{2\pi i \frac{n}{T}x} \tag{3}$$

Note: in these equations, the function $f(x)$ is represented as a series of cosines and sines along the interval $x \in [-\frac{T}{2}, \frac{T}{2}]$. Where T is the length of the domain of the function.

The measured signal is then projected onto each basis vector as in equation 2. This allows the signal to be represented as a simple sum of complex exponentials, where c_n is the length of the nth vector projection onto the basis vector, and n is the index of the basis vector. Thus, the measured signal can be represented as equation 3 [2]. This is important as each basis vector already has a set frequency which makes it favourable over a polynomial basis. The Fast Fourier Transform can remove noise seen in figure 5 by filtering out higher frequency waves. A signal made up of simple frequencies in figure 6 is left behind which is easier to analyse.

B. Data Science

B.1. Machine Learning

There are two main types of machine learning methods: supervised and unsupervised [17]. A supervised learning method refers to the existence of ground-truth labels with the collected data which the algorithm learns from. Unsupervised learning refers to the use of unlabelled data in building the algorithm. Unsupervised learning methods can be more unpredictable but be able to handle more complex processing tasks than supervised learning methods. For our project, we will only be implementing the most appropriate supervised machine learning method.

B.2. Random Forest

Random forest is an ensemble learning method for classification that constructs multiple decision trees which, when combined, lead to a final predictive output [9]. Random forest creates as many trees on the subset of the data and combines the output of all the trees. In this way, it reduces overfitting problems in decision trees and also reduces the variance, and therefore improves the accuracy.

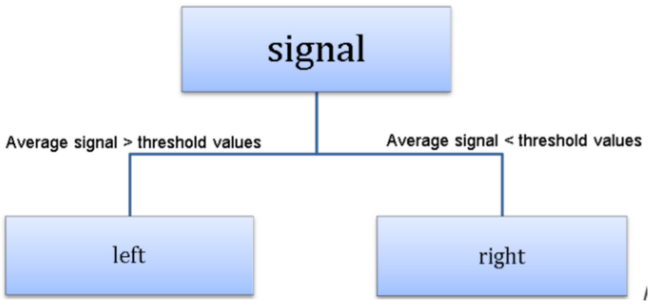

Fig. 7. A simplified example of a decision tree.

B.3. SVM Method

Support Vector Machine (SVM) is a supervised classification method which can be used for both classification or regression, mostly used in classification problems [10]. SVM method separates data by having a decision boundary between any two classes to classify them (hyperplanes).

B.4. KNN Method

K-Nearest Neighbors algorithm (KNN) is a non-parametric (no assumption on data distribution) learning algorithm [10]. KNN methods classify a data point by a majority of its K nearest neighbors. If K = 1, then the point is classified as the class of its nearest neighbor.

B.5. Features

Feature is a predictor variable which the model uses to understand whether the eye movement was to the left or to the right [10]. The performance of the model will be focused on the accuracy of the predicted sequence of Lefts and Rights.

4. METHODOLOGY

A. Raw Data Collection

The Spikerbox (See Section 3.A.1.) acts as an electrooculogram (EOG) to record eye movement signals and produce raw .wav files for the Data Science team to analyse. The recorded signals were measured within 5-15 Hz. Additionally, we manually labelled the ground truth within .txt files to supplement the .wav file which is essential to determine the true accuracy of our product. The ground truth is the information provided by direct observation during the recording process as opposed to information provided by inference. The ground truth labels we used were left eye movements as 1, right eye movements as 0, and a hard blink as 5, each with corresponding time stamps. A hard blink is an exaggerated blinking motion which can be characterised by a huge spike in the signal's amplitude. The timestamps in the ground truth allows us to detect the existence of false positives (signals detected by the code that did not occur).

Our Physics team generated over 30+ of such .wav and .txt file combinations which consisted of blinking and moving the eyes left and right. Each file recording captured different patterns of eye movements such as moving one's eye at irregular time intervals to mimic the different thinking times a player may require for each question. Throughout all recordings, regular blinking occurred, so as to be distinguished from hard blinks (which we will utilise for skips). This is important for the preliminary data cleaning process.

B. Raw Data Cleaning

For the real-time deployment of our Trivia game, there are a number of issues to resolve in regards to data-cleaning. This includes:

- Unwanted noise being detected from the Spikerbox
- Random error induced from changing the position of the electrodes
- Different people, due to physiological differences, registering different amplitudes for a given eye movement

As such, there are a number of methods we applied to clean the incoming data. Firstly, we used the Fast Fourier Transform as a low pass filter. By parsing frequency components of input signals, this filtered out any unwanted noise above 15 Hz.

Furthermore, in an attempt to minimise variation in detected signal amplitude between users, we normalised the amplitude height of the incoming signal to give a consistent amplitude bound of approximately [-1 , 1]. To maximise the cleanliness of the signal, we altered the game such that there is a 4 second pause between questions appearing where the user is forced to not move their eyes. This creates an opportunity for the signal to adjust back to 0.

B.1. Machine Learning Method

Machine learning examines the unique features of a data set and makes predictions about future observations with incomplete information. There are a variety of different machine learning methods and it is imperative we select the one most appropriate for our purposes. In "machine-learning language", we essentially have a classification problem: when given a signal input we need to classify whether it is a left movement, right movement or a blink.

Fortunately, the Physics team provided labelled data-sets to be used for supervised machine learning. A supervised learning technique refers to our model preparation process from which the model will "learn" from already existing correct labels. Now given we have only 3 categories of events, we will not consider clustering algorithms such as k-means clustering. Our analysis suggests that the most appropriate options include Instance-based Algorithms such as k-Nearest Neighbour (KNN) and Support Vector Machines (SVM) or an Ensemble-Learning algorithm such as the Random Forest (RF) method [12]. Based on the experience and research of the Data Science team however, our team decided on applying the RF method for three key reasons:

- RF functions more efficiently with larger data-sets which we will be expecting under live-streaming conditions as people play the game. It also works more effectively with a shorter pre-processing time [13]
- RF is inherently multi-class and provides a stronger platform for future growth when we wish to determine different eye movements - whereas other Instance-based methods are more intricate.
- There is no significant difference in accuracy for RF or SVM for these types of problems. Our Data Science team however has found that RF typically produces a slightly higher accuracy - which is the main performance metric for our product.

To apply the RF method, we take advantage of the machine learning scikit-learn python package. Specifically, we leverage the RandomForestRegressor function, which fits a number of classifying decision trees (See 3.B.2 for full description of Random Forest method).

B.2. Threshold Method

The Thresholding method takes a simpler approach: where a result is recorded whenever a threshold is touched in a specific time window. There is a different threshold level for each predicted outcome (i.e. Left, Right and Blinks). The complication is less of how to construct it - but more of where to set the threshold.

To determine the threshold, we used a highly computational technique. A normalised calibration signal was examined using a simple filtering system using a series of "if-statements" - which served to identify events based on a threshold. This technique is looped over a vector of potential threshold values and a range of thresholds that allows the detection of a predetermined sequence with 100% accuracy. The best threshold level is one that detects the real events most accurately and is high enough to ignore false positives. We chose the 80th percentile threshold level within that range. This is high to filter out most random fluctuations that are not events but it is not high enough to ignore actual events such as 'left' and 'right'.

Thus, an iterative process on the calibration signal is used to arrive at an appropriate threshold level, independent of human bias. This threshold level would change depending on the person and their calibration signal. However, on average, the threshold levels would be similar to that shown below:

- Left threshold $\approx$ 0.35
- Right threshold $\approx$ -0.35
- Blinks threshold = 1.3

C. Model Evaluation

The issue of model evaluation is quite nuanced. There are numerous different performance metrics to determine whether a model is "optimal" - such as Recall, Precision and the F1 Score to name a few. However, we believe that this is no simple machine-learning problem and the blind application of a confusion matrix is not appropriate here. This is because a key requirement in producing a confusion matrix is that the array of predicted events matches the array of ground-truth labelled events in length. However this is extremely difficult to control in our context, while maximising the flexibility of our algorithm to make predictions.

For example, let's say our user looks once left followed by once right. Our ground-truth labels are [L,R]. But let's say our algorithm predicts an extra event where there isn't any - for example [L,R,R]. This will prevent our analysis via a confusion matrix like in figure 8. Instead, we need a method of evaluating the model even if there are additional incorrect events detected.

		Ground Truth	
Predicted		Left (1)	Right (0)
	Left (1)	True Positives (TPs)	False Positives (FPs)
	Right (0)	False Negatives (FNs)	True Negatives (TNs)

Fig. 8. A schematic of a confusion matrix.

Hence, the difflab package was used to explore the accuracy of both the Machine Learning method and Threshold method results.

D. Deployment

To deploy the product, we created a simple user interface in Jupyter Notebook as seen in figure 9. This user interface includes:

- A question bank with 40+ questions so we never run out of questions with a randomised order each time
- A timer counting down from 120 seconds to ensure the game actually ends
- A set of "print" statements communicating to the user the instructions of the game as well as their score
- Another timer between the detection of events so the user's eyes stop moving

5. RESULTS

The threshold method has an average accuracy of 93.53% and when hard blinks are introduced, an average accuracy of 92.53% as seen in figure 10. This demonstrates the versatility and robustness of the threshold method. Table 1 also demonstrates how the accuracy over subsequent trials also does not decay very much and remains at around 90%.

If we contrast this accuracy to the results found in figure 11, we see a dramatic decrease in accuracy. Here, the median accuracy is below 90% and accuracy falls to around 40% for some sequences. This box plot is a little misleading. If we look at table 2, we see that a decently high accuracy is achieved for the first six sequences. However, over time the accuracy of detection decays.

Hence, table 2 demonstrates that the Random Forest method is viable method of signal detection. However, it lacks the same accuracy and robustness over extended periods of times that the threshold method has.

Different User Data

To provide further results for the threshold method, a different user played our trivia quiz. Table 3 shows the accuracy for 4 sequences.

For sequence 3, the code was able to detect a single hard blink which coincided with the timing of real life. However, the two hard blinks in sequence 4 were undetected.

```
Left = True, Right = False

1 The Titanic sank in 1913.
You said True, that is Incorrect
The Titanic sank in 1913. False

2 World War II ended in 1945.
You said True, that is correct
World War II ended in 1945. True

3 There have been 50 U.S. Presidents.
You said False, that is correct
There have been 50 U.S. Presidents. False

4 Queen Elizabeth II was born in 1926.
You said True, that is correct
Queen Elizabeth II was born in 1926. True

5 Saint Patrick's Day was originally associated with color red.
You said False, that is correct
Saint Patrick's Day was originally associated with color red. False

6 Zimbabwe was known as Rhodesia from 1964 to 1980.
You skipped
Zimbabwe was known as Rhodesia from 1964 to 1980. True

7 The Cold war officially ended in 1979.
You said True, that is Incorrect
The Cold war officially ended in 1979. False

8 Christopher Columbus discovered the New World in 1592.
You said False, that is correct
Christopher Columbus discovered the New World in 1592. False

9 The name for the Greek goddess of victory is called Nike.
You said False, that is incorrect
The name for the Greek goddess of victory is called Nike. True

10 Leonardo Da Vinci painted the late 15th century mural known as Last Supper.
You said True, that is correct
Leonardo Da Vinci painted the late 15th century mural known as Last Supper. True

11 The Titanic sank in 1913.
Accuracy = 0.9
[ 1.    4.25  1.    8.15  0.   10.9   1.   15.35  0.   18.4   5.   21.75
  1.   25.15  0.   28.55  0.   32.2   1.   36.45]
Time is up! Your Score: 6 Out of 9 , Skipped: 1
```

Fig. 9. An example of the deployment of the trivia quiz. Note: this is a simulation of the live process, using pre-recorded files. Thus, a smaller cooldown between detected events was used.

Further analysis can be observed in the individual discussion.

6. RESULTS DISCUSSION

A. The Live Product

There are shortcomings within the live product itself. Unlike the simulated live version seen in figure 9, the cool down period needs to be much longer. This is a consequence of the code itself. Between questions, a 4 second timer was implemented between left and right movement detection and the next question reappearing. This is because, if this time is shortened, the signal may not have enough time to readjust itself to around 0. Thus, a false positive or event may be detected despite the intent of the user in this time period.

When considering hard blinks, this signal is very large in amplitude. An even longer time period to readjust the measured amplitude signal to 0 is required. An 8 second cool down was needed to ensure that no false positives were detected during that 8 second time period between a new question appearing. For this reason, the product was designed to be unable to detect consecutive blink events. If a user tries to input 2 consecutive blink events, then the second blink will be detected as a left or right event instead. Consecutive blink events are discentivised as it stagnates play time.

B. Importance of Calibration

Calibration is integral to the success of the product. The Spikerbox will detect different signal strengths for different people due to physiological variation or the amount of electrode gel used. In addition, how each person positions their headband will change the location of electrodes.

The live code produced a signal output which averaged between the range of 500-700 arbitrary units. This average would fluctuate over time even with the same individual. Hence, after a period of time ($\approx 10\ mins$), the user must re-calibrate input signal. An average calibration signal was used to centre the output signal to zero. This signal would then be normalised by dividing the entire signal over the maximum variation from the calibration signal's average. Hence, a common threshold between users can be used to detect 'left', 'right', and 'blink' events.

The threshold value determined in this way will allow users to blink and relax the usage of eyes during the calibration run. Small blinks are encouraged as they are normal and will register as a slight bump in the signal which could be considered a false positive if it reaches the arbitrary threshold.

The iterative process to determine threshold level is indepen-

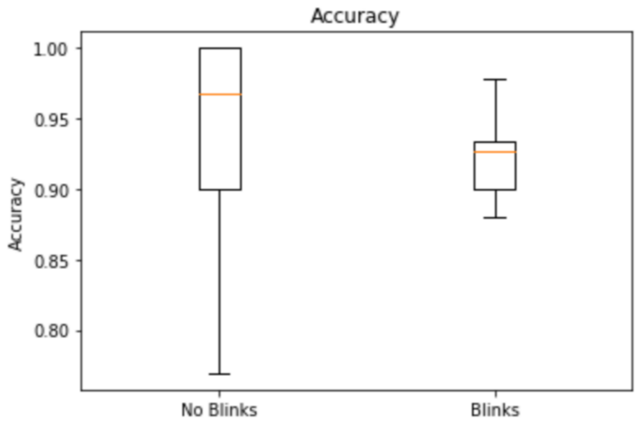

Fig. 10. Average accuracy without hard blinks was 93.53%, while with hard blinks an average accuracy of 92.53% was achieved.

Sequence A	No Blinks %	Sequence B	Blinks %
1	100.0	1	90.00
2	100.0	2	88.00
3	100.0	3	92.68
4	100.0	4	96.30
5	87.18	5	97.78
6	90.91	6	89.66
7	96.77	7	92.75
8	90.00	8	93.33
9	76.92	9	92.31

Table 1: Accuracy of no blink and blink sequences over 9 trials for the threshold method. Sequence A and sequence B are separate sequences and they are not the same for a given index.

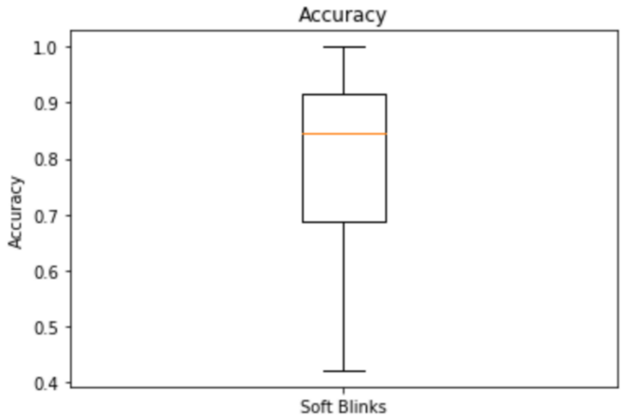

Fig. 11. Accuracy for the random forest method.

Sequence	No Blinks %
1	92.86
2	91.67
3	84.62
4	100.0
5	89.47
6	82.35
7	68.75
8	42.11
9	48.00

Table 2: Accuracy of no blink sequences over 9 trials for the Random Forest method.

dent of human input. Even with different users, a relatively high accuracy can be observed in tables 1 and 3. Thus, the product has demonstrated the ability to determine a 'correct' threshold value despite different users.

It is also important to note that the median accuracy 84.62% for the Random Forest method in figure 11 and table 2 may be overstated. This is because this method currently uses human input to determine the correct threshold to use. Thus, to improve the Random Forest method, the threshold level should be dependent on the calibration signal. This would remove human bias in determining the true accuracy and utility of the Random Forest method in respect to different users.

C. Importance of Equipment

The consistency of the Spikerbox proved to be an issue for this project. Almost randomly at times, it would not work and this is unacceptable for a commercial product. If we were to further develop this technology, we will seek out better equipment to detect the brain signals. Further, if we wish to expand the detection capacity of the technology we will also need additional electrodes.

Sequence	Accuracy %
1	90.00
2	80.00
3	90.00
4	57.14

Table 3: Results for four sequences this user gave us. It is important to note that sequences 3 and 4 include hard blinks, whilst sequences 1 and 2 do not have any hard blinks.

D. Shortcomings with Machine Learning

Applying machine learning to our project has several shortcomings. Firstly, we recognised that as the game ran for longer - and so more data was produced - the accuracy became lower and lower. We analysed the reason for this to be that simply as the user played for longer - the algorithm is more likely to encounter random events or complex signals it has not previously learned. Thus, this points to the lack of training set data for our algorithm to learn from. Our Data Science team suggests however, that machine learning is still very applicable. The fact is that we simply don't have the resources to generate enough data, ideally from multiple people, to very comprehensively train the algorithm such that it can perform.

To improve the robustness of the Random Forest method, a longer calibration signal needs to be run. This would allow for a better determination of the true signal from the user. However, this would sacrifice usability as there is a longer startup time before the game can be run. In addition, the calibration run is only effective if the algorithm detects exactly the same amount of signals as a predetermined sequence that the user must run. For instance, if the sequence fed into the algorithm has 20 samples, unless exactly 20 samples (left or right signals) is detected, then the Random Forest method will not work. Thus, by introducing a longer required calibration run, it decreases the usability of the product. To provide perspective, the calibration signal for the threshold method only required 8 samples and has a duration of $< 18\ s$. The Random Forest method used a calibration signal which had 14 samples and ran for a duration of $< 25\ s$. There is also no ability to speed up the calibration process as if time between events is not large enough, then the detection process will be inaccurate. Given that the live version of the product requires longer cooldowns, a longer calibration length means that calibration signals to improve accuracy could take as long as a minute, and even then, it may not be a valid calibration signal due to difference in sequence length when compared to the predetermined sequence.

Due to short calibration length required, and stability over time, the threshold method is superior as it boasts a higher accuracy as seen figure 10 and table 1. When considering the deployment in a real life scenario where calibration cannot be rushed, it is also the more user friendly choice.

E. How Accuracy was Determined

Since the detected sequence length often differed from the ground truth sequence length, our team explored the use of two packages:

- StringMatcher function from the fuzzywuzzy package

- SequenceMatcher function from the difflab package

The fuzzywuzzy package utilises the Levenshtein Distance - which in simple terms measures "the minimum number of edits that you need to change a sequence to another". These edits can be insertions, deletions or substitutions. Here is a formal definition:

The Levenshtein distance between two strings a, b (of length $|a|$ and $|b|$ respectively) is given by $lev_{a,b}(|a|,|b|)$ in equation 4:

$$lev_{a,b}(i,j) = \begin{cases} max(i,j), \text{ if } min(i,j) = 0 \\ min \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \text{ otherwise.} \\ lev_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} \end{cases} \tag{4}$$

where $1_{(a_i \neq b_j)}$ is the indicator function equal to 0 when $a_i = b_j$ and equal to 1, otherwise, and $lev_{a,b}(i,j)$ is the distance between the first i characters of a and the first j characters of b. i and j are 1-based indices.

Note that the first element in the minimum corresponds to deletion (from a to b), the second to insertion and the third to match or mismatch, depending on whether the respective symbols are the same [6].

On the other hand, the difflab package analyses the sequence from left to right - locating the maximum "matching blocks" of string. The function primarily examines the number of triples that are similar and outputs a ratio through division by the total number of elements.

For our purposes, the fuzzywuzzy package using the Levenshtein distance over-estimates how accurate the prediction actually is. For example 1, the fuzzywuzzy package would estimate the accuracy to be 0.67. It does not punish wrong predictions fairly - as the final output will be quite similar to the actual signal given that there are only three options to choose from.

```
from fuzzywuzzy import fuzz, stringMatcher
import difflib
# As long as python-Levenshtein is available, that will be used for follwing:
fuzz.ratio("ababab", "aaaaab")
#67
# Switch to difflib:
fuzz.SequenceMatcher = difflib.SequenceMatcher
fuzz.ratio("ababab", "aaaaab")
#33
```

Example 1: How to measure accuracy of sequences.

The similarity ratio from the difflab package [14] can be seen in equation 5:

$$\text{Ratio} = \frac{2 \times \text{length of same sequence}}{\text{Sum of sequence lengths}} \tag{5}$$

$$\begin{aligned} \text{Ratio in example 1} &= \frac{2 \times \text{length of } `ab' \ (\underline{ab}abab, aaaa\underline{ab})}{\text{Sum of sequence lengths}} \\ &= \frac{2 \times 2}{12} \\ &= 0.33 \end{aligned}$$

Thus, SequenceMatcher from the difflab package was used to measure accuracy.

F. Possible Expansions of the Product

To improve the accuracy of both detection methods, a method that filters out false positives better is required. As someone reads a question on the trivia quiz, their eyes will move from left to right, generating a small signal. However, if during this time the user blinks as they normally would, this would also generate a signal with an amplitude usually below 0.1 as seen in figure 12. When these two factors are compounded together in addition to other circumstances, the strength of the signal may be large enough to reach the threshold level.

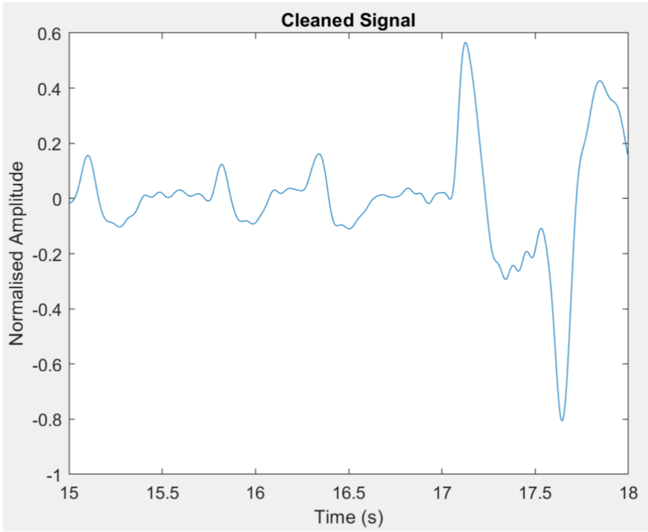

Fig. 12. 3 small blinks before a left look on a cleaned and normalised signal.

What we propose is the enhancement of the simple detection method. Rather than looking for a signal that reaches one threshold, we look for a signal that touches a sequence of thresholds as seen in table 4. When we look at this method and observe the graph in figure 12, we infer that this method would serve as a better filtering system for small blinks as it adds more requirements for an event to be characterised as 'left' or 'right'.

This leads to our next proposition to expand the product itself. Wu, S. et al. has already been able to detect events in eight directions with an average classification accuracy of 88.59% [5]. While this offers the users more options, it demonstrates a lower

Digital Value / Directions	Vertical Sequence	Horizontal Sequence
Up	0121(-1)(-2)(-1)	0000000
Down	0(-1)(-2)(-1)121	0000000
Left	0000000	0(-1)(-2)(-1)121
Right	0000000	0121(-1)(-2)(-1)
Up-right	0121(-1)(-2)(-1)	0121(-1)(-2)(-1)
Up-left	0121(-1)(-2)(-1)	0(-1)(-2)(-1)121
Down-right	0(-1)(-2)(-1)121	0121(-1)(-2)(-1)
Down-left	0(-1)(-2)(-1)121	0(-1)(-2)(-1)121

Table 4: The categorisation of the movement of eyes in eight directions is done by attributing a sequence to each direction [5].

User / Direction	S1	S2	S3	S4	S5	S6	S7	S8	Total	STD
Up	100	100	100	100	90	90	90	100	96.25	5.1755
Down	100	100	100	100	100	80	90	100	96.25	7.4402
Left	70	90	100	90	100	100	100	90	92.50	10.351
Right	90	100	90	100	90	80	80	90	90.00	7.5593
Up -right	60	90	90	80	70	100	100	80	83.75	14.0789
Up -left	80	90	100	70	60	100	80	100	85.00	15.1186
Down -right	70	100	80	50	80	90	80	80	78.75	14.5774
Down -left	80	100	100	70	90	70	90	90	86.25	11.8773
Total	81.25	96.25	95.00	82.50	85.00	88.75	88.75	91.25	88.59	

Table 5: The accuracy of classification for eight users in eight directions [5].

accuracy than what is seen in figure 10. It was determined that the detection in normal directions was more stable than oblique directions. Their accuracy of classification with eight users was above 90% for 'left', 'right', 'up', and 'down'. This can be seen in table 5 as being comparable to our own accuracy in figure 10 whilst the oblique directions offered an accuracy below 88%. The experimental setup they used can be seen in figure 13. Offering four directions (up, down, left, right) to our product seems to be a good balance between accuracy and enhanced functionality.

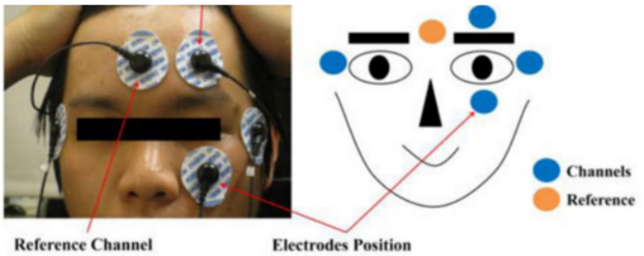

Fig. 13. Placement of electrodes used in Wu, S. et al. [5]

Overall, applying machine learning to our project had several shortcomings. As the game ran for longer - and so more data was produced - the accuracy became lower and lower. We analysed the reason for this to be that simply as the user played for longer - the algorithm is more likely to encounter random events or complex signals it has not previously learned as conditions continue to diverge from the time of calibration. This points to the lack of training data for our algorithm to learn from. Our Data Science team suggests, however, that machine learning is still very applicable. The fact is that we simply do not have the resources to generate enough data, ideally from multiple people, to very comprehensively train the algorithm such that it can perform. Ultimately, it is for this reason that the Thresholding method reigned superior.

7. CONCLUSION

Our product enables people to play a quiz that can be answered with changes in the eye movements and blinking only. This brings a huge benefit to society as it is easy to play and allows physically disadvantaged people who cannot participate in physically demanding games to also enjoy. To make this product work, we have collected data using a Spikerbox which relies on the concept of our eyes as dipoles and cleaned electrical brain signals by a Fast Fourier transform. We then built a classifier using random forest, ran simulations over the collected data and determined accuracy. A question bank of trivia question was also made and connected to the classifier. Through understanding the human eye, we used a combination of Physics and Data Science concepts to detect signals from the eye and translate it into inputs to our game. We came to the conclusion that the threshold method of event detection was more accurate than the machine learning method, given the highly limited access to quality training data.

8. STUDENT CONTRIBUTIONS

Jeffrey Zheng (480345040)

Jeffrey collaborated closely with Max and Ming to analyse the accuracy of their classification methods using the SequenceMatcher function from difflab to decide which should be implemented for the live product. Additionally, worked on the presentations and helped make things look aesthetic in general. Also, allocating roles for the report.

Michael Lin (470000393)

Michael collaborated closely with the physics team to help to convert the coding from MatLab and Rstudio to python. Moreover, he had improved the coding of the quiz function such as adding a timer countdown during the quiz and speed up the process of user input. Furthermore, he had built a test version of the question bank and quiz function for the testing by tutor Alison.

Linzhi Jiang (480036621)

Lingzhi collaborated closely with the physics group. He generated the question bank for Max and Tyrone for the actual quiz. Moreover, Lingzhi has taken charge of writing down all of the meeting notes since mid semester break as well as combining them with the physics one. Besides, he also helped interpret the result of the accuracy.

Haeri Min (480401391)

Haeri worked as part of the data group to write code that can detect hard blinks and tested a sample threshold to observe accuracy for soft blinks using the group's pre-recorded data. She also researched the benefits of using random forest and how effective the machine learning process is as a part of the report.

Mingjie Zhang (470132029)

Leading the design of Machine Learning and creating the main functions of it. For example, event detect function, normalize function. In order to make random forest more readable, reusable and suitable for our project, random forest function has been established that is easy for us to adjust and modify. Cooperated with Haeri to complete the judgment and pre-treatment of hard blink. In charge of the progress of the physics and data group. Communicated well throughout and was flexible with scheduling. Helped the physics and data group deal with problems and code errors whilst taking initiative in pioneering the Machine Learning Method.

Tyrone Simbul (460379786)

Tyrone worked with Max as part of the physics group to conduct the initial data collection process. He also wrote the initial code for the question bank and quiz with the help of physics tutor Alison. This initial code did not incorporate eye movement detection. He made preliminary observations of accuracy in the testing of thresholds for the threshold method using the group's pre-recorded data. Finally, he composed the first draft of the raw data collection and processing sections of the final report.

Max Xiao (480377535)

Max was involved in all parts of the project and was responsible for combining everyone's work. He had the Spikerbox and thus, took the role of getting a product that worked. Wrote the thresholding method that was used throughout the project, and was able to be versatile. Moved the entire project into a simulated version once the Spikerbox started having some issues with consistency. Made the videos for presentation. Also analysed and reviewed accuracy results. Made a working live product.

REFERENCES

1. Anthony H. V. Schapira, Edward Bryne, Richard S. J. Frackowiak, Richard T. Johnson, Yoshikuni Mizuno, Martin A. Samuels, Stephen D. Silberstein, Zbigniew K. Wszolek. Neurology and Clinical Neuroscience (2007). *Section V Neuro-Otology: pp. 234*. Philadelphia, Mosby Elsevier.
2. H. C. Taneja. Advanced Engineering Mathematics (2010). *Volume II: pp. 232-234*. New Delhi, I.K International Publishing House Pvt. Ltd.
3. Fuming Fang, Takahiro Shinozaki. Electrooculography-based continuous eye-writing recognition system for efficient assistive communication systems (2018). *PLoS ONE: 13*.
4. Paul A. Constable, Michael Bach, Laura J. Frishman, Brett G. Jeffrey, Anthony G. Robson. ISCEV Standard for clinical electro-oculography (2017). *Documenta Ophthalmologica. Advances in Ophthalmology: 134, 1-9*.
5. Shang-Lin Wu, Lun-De Liao, Shao-Wei Lu, Wei-Ling Jiang, Shi-An Chen, and Chin-Teng Lin. Controlling a Human–Computer Interface System With a Novel Classification Method that Uses Electrooculography Signals (2013). *Ieee Transactions On Biomedical Engineering: 60, 2133-2141*.
6. 'Levenshtein Distance,' Wikipedia, last modified: 20 May 2020. Retrieved from https://en.wikipedia.org/wiki/Levenshtein_distance.
7. Moler, C. Levenshtein Edit Distance Between Strings [Blog] (2017. Retrieved from https://blogs.mathworks.com/cleve/2017/08/14/levenshtein-edit-distance-between-strings/.
8. Arias, F. Fuzzy String Matching in Python (2017). Retrieved from https://www.datacamp.com/community/tutorials/fuzzy-string-python.
9. Breiman, L. Random Forests (2001). *Machine Learning, 45, 5-32*.
10. Kevin P. Murphy. Machine Learning A Probabilistic Perspective (2012). *pp. 2-10, 496-512*. Massachusetts, The MIT Press.
11. Guerzhoy, M. Nearest Neighbour Classifiers [Ebook] (2017). Retrieved from https://www.cs.toronto.edu/~guerzhoy/411/lec/W01/knn.pdf.
12. Li, H. Which machine learning algorithm should I use? [Blog] (2017). Retrieved from https://blogs.sas.com/content/subconsciousmusings/2017/04/12/machine-learning-algorithm-use/.
13. Random forest vs SVM. (2018). Retrieved from https://www.thekerneltrip.com/statistics/random-forest-vs-svm/.
14. theodickson. Difflib and python-Levenshtein give different ratios in some cases (2016). Retrieved from: https://github.com/seatgeek/fuzzywuzzy/issues/128.
15. University of Sydney. PHYS3888 + DATA3888 Week 1 - Project overview (2020) [Powerpoint].
16. Abdelkader N. Belkacem, Supat Saetia, Kalanyu Zintus-art, Duk Shin, Hiroyuki Kambara, Natsue Yoshimura, Nasreddine Berrached, and Yasuharu Koike. Real-Time Control of a Video Game Using Eye Movements and Two Temporal EEG Sensors (2015). *Computational Intelligence and Neuroscience: 2015, 1-10*.
17. Zhao, Z. and Liu, H. Spectral feature selection for supervised and unsupervised learning (2007). *Proceedings of the 24th international conference on Machine learning - ICML '07*.

9. APPENDIX

References for Questions:

- https://conversationstartersworld.com/space-trivia-questions/
- https://www.triviaquestionsnow.com/for/history-trivia
- https://trivia.fyi/category/computer-trivia/

Trivia Questions

Question	Truth Value
The universe is 13.8 billion years old.	TRUE
The constellation Virgo is the hottest place in the universe.	TRUE
Jupiter is the largest planet in our solar system.	TRUE
Apollo 17 was the last NASA manned mission space flight to the moon.	TRUE
There is only one moon in our solar system.	FALSE
The shortest space flight was 25 minutes.	FALSE
Light cannot escape from a black hole.	TRUE
There are 9 planets in the solar system.	FALSE
Red supergiant stars are the largest type of stars in the universe.	TRUE
Elliptical galaxies are not the most common galaxies in the universe.	FALSE
An ant can fit 100 times its own weight.	FALSE
8128 is a perfect number.	TRUE
There are 24 time zones in the world.	TRUE
AB positive is the rarest blood type in humans.	FALSE
An octopus has only one heart.	FALSE
Hydrogen is the first element on the periodic table.	TRUE
There are seven colors in a rainbow.	TRUE
Dolphins cannot smell.	TRUE
Liver is the largest organ of human body.	FALSE
Celsius and Fahrenheit are equal at -40 degrees.	TRUE
The Titanic sank in 1913.	FALSE
World War II ended in 1945.	TRUE
There have been 50 U.S. Presidents.	FALSE
Queen Elizabeth II was born in 1926.	TRUE
Saint Patrick's Day was originally associated with color red.	FALSE
Zimbabwe was known as Rhodesia from 1964 to 1980.	TRUE
The Cold war officially ended in 1979.	FALSE
Christopher Columbus discovered the New World in 1592.	FALSE
The name for the Greek goddess of victory is called Nike.	TRUE
Leonardo Da Vinci painted the late 15th century mural known as Last Supper.	TRUE
HP and Microsoft were all started in a Garage.	TRUE
Charles Babbage invented the first mechanical computer.	TRUE
GUI stands for Graphical user interface in computer science.	TRUE
Terabyte is the largest unit in storage.	FALSE
Apple was the first publicly traded company to reach a 1 trillion dollar market cap.	TRUE
ROM stands for read only memory in computer memory.	TRUE
SRGB has a wider range of colors than AdobeRGB.	FALSE
SQL stands for Selected Query Language.	FALSE
The first iphone was released in 2008.	FALSE
444 is the three digit when error code for censored content.	FALSE

Videos of the Product

- 120s Live, Normal Speed
- Different User Quiz 2, Normal Speed
- Different User Quiz 3, Normal Speed

Google Drive of Data and Code